PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-021830

(43)Date of publication of application : 28.01.1994

(51)Int.Cl. H03M 7/40
H04N 5/92
H04N 7/13

(21)Application number : 04-196218

(22)Date of filing : 30.06.1992

(71)Applicant : SONY CORP

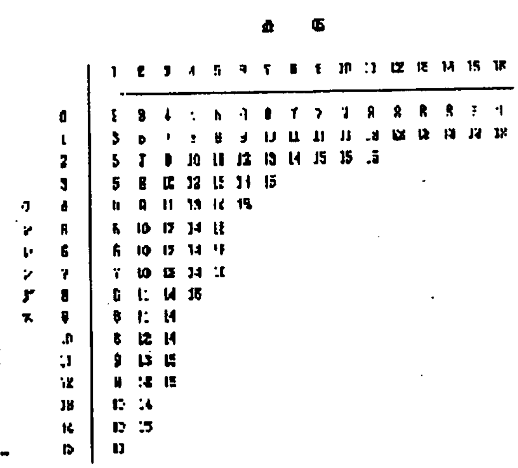
(72)Inventor : YANAGIHARA HISAFUMI
KURISU SHIYUU

(54) TWO-DIMENSION HUFFMAN CODING METHOD

(57)Abstract:

PURPOSE: To reduce a maximum bit length caused in the 2-dimension Huffman coding, to simplify the hardware and to generate a coding output easy for self-synchronization.

CONSTITUTION: A 16-bit maximum (including a sign bit) code word is generated after referencing a table based on a set of an amplitude and a run length. Bit allocation of the table is implemented by taking an incidence probability and quantization of DCT coefficient data into account. Furthermore, a generated code word including '1110' for a synchronization pattern as a final 4-bit pattern is in existence. First and 2nd escape sequences are applied to a data word at the outside of the table. In the escape sequence in the case of run length =0, a coded output whose total bit number is 16 is generated and in the escape sequence in the case of run length ≠0, a coded output whose total bit number is 31 is generated. The 31-bit data are sent while being divided into 15-bit data and 16-bit data.



LEGAL STATUS

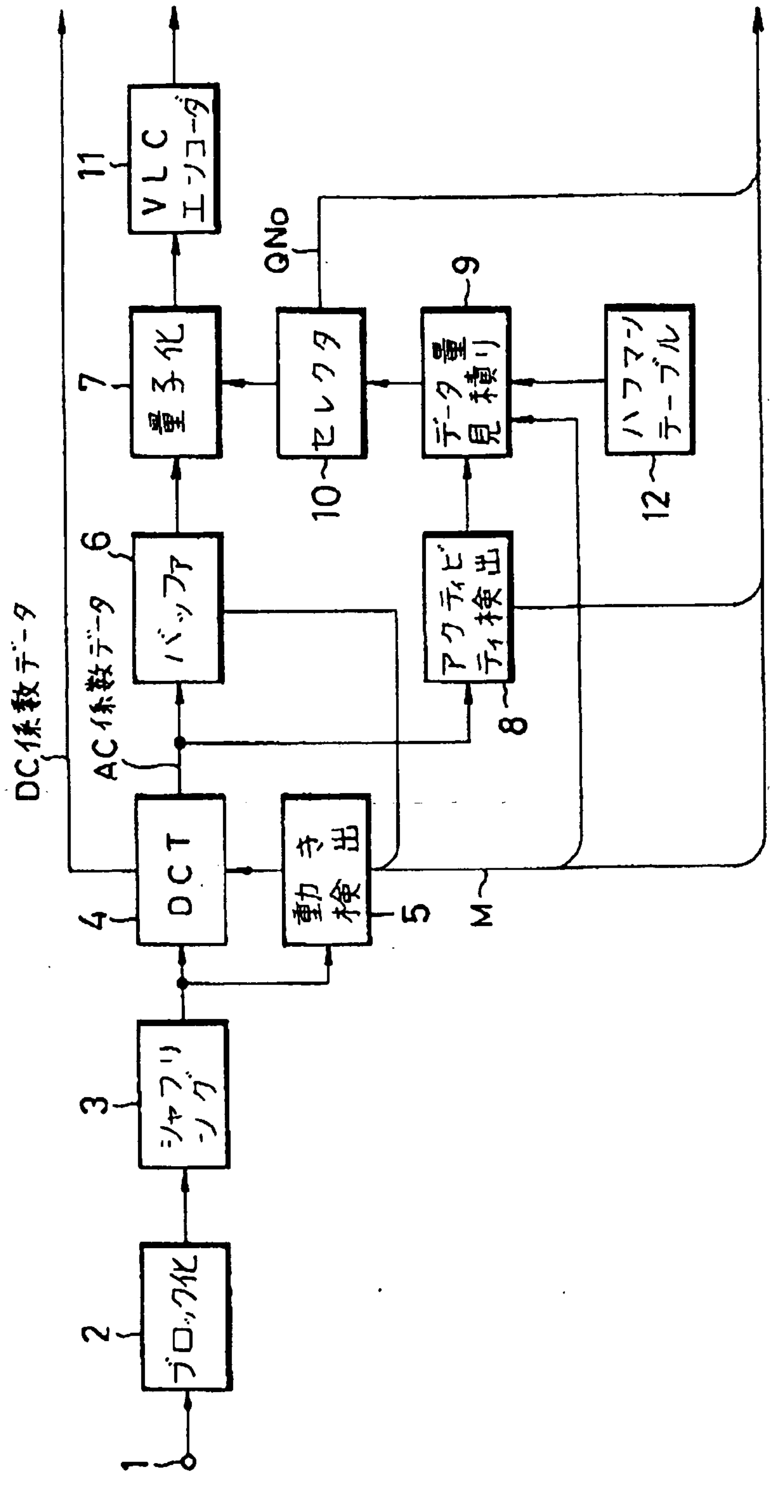
[Date of request for examination] 01.06.1999

[Date of sending the examiner's decision of rejection]	05.12.2000
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	
[Date of registration]	
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

* NOTICES *

DRAWINGS

[Drawing 1]


1
2
ブロック化
3
シャフリング
4
DCT
5
動き検出
M
DC係数データ
AC係数データ
6
バッファ
7
量子化
8
アクティビティ検出
9
データ量見積り
10
セレクタ
QNo
11
VLC
エンコーダ
12
ハフマンテーブル

[Drawing 2]

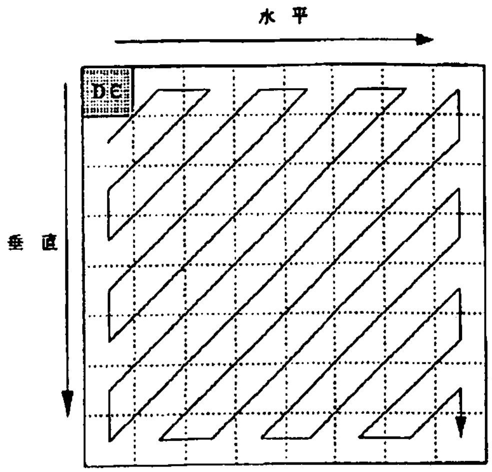
振幅

ランレングス		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	0	2	3	4	5	5	6	6	7	7	7	8	8	8	8	9	9
	1	3	5	7	8	9	9	10	11	11	11	12	12	13	13	13	13
	2	5	7	9	10	11	12	13	14	15	15	15					
	3	5	8	10	12	13	14	15									
	4	6	9	11	13	14	15										
	5	6	10	12	14	15											
	6	6	10	12	14	15											
	7	7	10	13	14	15											
	8	8	11	14	15												
	9	8	11	14													
	10	8	12	14													
	11	9	13	15													
	12	9	13	15													
	13	10	14														
	14	10	15														
	15	11															

[Drawing 4]

111111111110s	sync	5	3	12
011111111110s	sync	3	4	12
001111111110s	sync	1	11	12
000111111110s	sync	1	12	12
000001111110s	sync	10	2	12
000000111110s	sync	2	6	12
000000001110s	sync	6	3	12
1111111111110s	sync	11	2	13
0111111111110s	sync	1	13	13
0011111111110s	sync	7	3	13
0001111111110s	sync	1	14	13
0000011111110s	sync	4	4	13
0000001111110s	sync	2	7	13
0000000011110s	sync	3	6	13
0000000001110s	sync	1	16	13
0000011000001s		12	2	13
0000010000001s		1	16	13
11111111111110s	sync	5	4	14
01111111111110s	sync	9	3	14
00111111111110s	sync	8	3	14
00011111111110s	sync	2	8	14
00000111111110s	sync	3	6	14
00000011111110s	sync	7	4	14
00000000111110s	sync	10	3	14
00000000011110s	sync	6	4	14
00000000001110s	sync	13	2	14
00000110000101s		4	5	14
111111111111110s	sync	3	7	15
011111111111110s	sync	11	3	15
001111111111110s	sync	6	5	15
000111111111110s	sync	2	9	15
000001111111110s	sync	4	6	15
000000111111110s	sync	8	4	15
000000001111110s	sync	14	2	15
000000000111110s	sync	7	5	15
000000000011110s	sync	6	5	15
000000000001110s	sync	12	3	15
000001100001101s		2	11	15
000001100001001s		2	10	15

[Drawing 10]



[Drawing 3]

Codeword	Type	Runlength	Amplitude	Code Length (bits)
10s		0	1	2
110s		0	2	3
010s		1	1	3
1110X	sync	EOB (End of Block)		4
0110s		0	3	4
11110s	sync	0	4	6
01110s	sync	2	1	6
00110s		0	5	6
00101s		1	2	6
00001s		3	1	6
111110s	sync	0	6	6
011110s	sync	4	1	6
001110s	sync	5	1	6
001001s		0	7	6
000110		ESC1 (Escape Code 1)		6
000101s		6	1	6
1111110s	sync	0	8	7
0111110s	sync	1	3	7
0011110s	sync	0	9	7
0001110s	sync	7	1	7
0010001s		0	10	7
0001001s		2	2	7
11111110s	sync	0	11	8
01111110s	sync	8	1	8
00111110s	sync	1	4	8
00011110s	sync	9	1	8
00100001s		0	12	8
00100000s		0	13	8
00010001s		3	2	8
00010000		ESC2 (Escape Code 2)		8
00000101s		0	14	8
00000001s		10	1	8
111111110s	sync	1	5	9
011111110s	sync	0	15	9
001111110s	sync	11	1	9
000111110s	sync	0	16	9
000001110s	sync	4	2	9
000001101s		12	1	9
000001001s		2	3	9
000000101s		1	6	9
1111111110s	sync	5	2	10
0111111110s	sync	13	1	10
0011111110s	sync	1	7	10
0001111110s	sync	6	2	10
0000011110s	sync	14	1	10
0000001110s	sync	3	3	10
0000011001s		2	4	10
0000010001s		7	2	10
11111111110s	sync	1	8	11
01111111110s	sync	1	9	11
00111111110s	sync	15	1	11
00011111110s	sync	4	3	11
00000111110s	sync	8	2	11
00000011110s	sync	9	2	11
00000110001s		1	10	11
00000100001s		2	5	11

[Drawing 5]

```
0000001    , 1, /*  */
0000010    , 2, /*  */
0000011    , 3, /*  */
0000100    , 4, /*  */
0000101    , 5, /*  */
0000110    , 6, /*  */
0001000    , 7, /*  */
0001001    , 8, /*  */
0001010    , 9, /*  */
0001011    , 10, /*  */
0001100    , 11, /*  */
0001101    , 12, /*  */
0010000    , 13, /*  */
0010001    , 14, /*  */
0010010    , 15, /*  */
0010011    , 16, /*  */
0010100    , 17, /*  */
0010101    , 18, /*  */
0010110    , 19, /*  */
0011000    , 20, /*  */
0011001    , 21, /*  */
0011010    , 22, /*  */
0011011    , 23, /*  */
0100000    , 24, /*  */
0100001    , 25, /*  */
0100010    , 26, /*  */
0100011    , 27, /*  */
0100100    , 28, /*  */
0100101    , 29, /*  */
0100110    , 30, /*  */
0101000    , 31, /*  */
0101001    , 32, /*  */
0101010    , 33, /*  */
0101011    , 34, /*  */
0101100    , 35, /*  */
0101101    , 36, /*  */
0110000    , 37, /*  */
0110001    , 38, /*  */
0110010    , 39, /*  */
0110011    , 40, /*  */
0110100    , 41, /*  */
0110101    , 42, /*  */
0110110    , 43, /*  */
1000000    , 44, /*  */
1000001    , 45, /*  */
1000010    , 46, /*  */
1000011    , 47, /*  */
1000100    , 48, /*  */
1000101    , 49, /*  */
1000110    , 50, /*  */
1001000    , 51, /*  */
1001001    , 52, /*  */
1001010    , 53, /*  */
1001011    , 54, /*  */
1001100    , 55, /*  */
1001101    , 56, /*  */
1010000    , 57, /*  */
1010001    , 58, /*  */
1010010    , 59, /*  */
1010011    , 60, /*  */
1010100    , 61, /*  */
1010101    , 62, /*  */
```

[Drawing 6]

```
000000001     , 1, /*      */
000000010     , 2, /*      */
000000100     , 3, /*      */
000000101     , 4, /*      */
000000110     , 5, /*      */
000001000     , 6, /*      */
000001001     , 7, /*      */
000001010     , 8, /*      */
000001100     , 9, /*      */
000001101     , 10, /*     */
000001110     , 11, /* sync */
000010000     , 12, /*     */
000010001     , 13, /*     */
000010010     , 14, /*     */
000010100     , 15, /*     */
000010101     , 16, /*     */
000010110     , 17, /*     */
000011000     , 18, /*     */
000011001     , 19, /*     */
000011010     , 20, /*     */
000011110     , 21, /* sync */
000100000     , 22, /*     */
000100001     , 23, /*     */
000100010     , 24, /*     */
000100100     , 25, /*     */
000100101     , 26, /*     */
000100110     , 27, /*     */
000101000     , 28, /*     */
000101001     , 29, /*     */
000101010     , 30, /*     */
000101100     , 31, /*     */
000101101     , 32, /*     */
000101110     , 33, /* sync */
000110000     , 34, /*     */
000110001     , 35, /*     */
000110010     , 36, /*     */
000110100     , 37, /*     */
000110101     , 38, /*     */
000110110     , 39, /*     */
000111110     , 40, /* sync */
001000000     , 41, /*     */
001000001     , 42, /*     */
001000010     , 43, /*     */
001000100     , 44, /*     */
001000101     , 45, /*     */
001000110     , 46, /*     */
001001000     , 47, /*     */
001001001     , 48, /*     */
001001010     , 49, /*     */
001001100     , 50, /*     */
001001101     , 51, /*     */
001001110     , 52, /* sync */
001010000     , 53, /*     */
001010001     , 54, /*     */
001010010     , 55, /*     */
001010100     , 56, /*     */
001010101     , 57, /*     */
001010110     , 58, /*     */
001011000     , 59, /*     */
001011001     , 60, /*     */
```

[Drawing 7]

```
001011010    , 61, /*  */
001011110    , 62, /* sync */
001100000    , 63, /*  */
001100001    , 64, /*  */
001100010    , 65, /*  */
001100100    , 66, /*  */
001100101    , 67, /*  */
001100110    , 68, /*  */
001101000    , 69, /*  */
001101001    , 70, /*  */
001101010    , 71, /*  */
001101100    , 72, /*  */
001101101    , 73, /*  */
001101110    , 74, /* sync */
001111110    , 75, /* sync */
010000000    , 76, /*  */
010000001    , 77, /*  */
010000010    , 78, /*  */
010000100    , 79, /*  */
010000101    , 80, /*  */
010000110    , 81, /*  */
010001000    , 82, /*  */
010001001    , 83, /*  */
010001010    , 84, /*  */
010001100    , 85, /*  */
010001101    , 86, /*  */
010001110    , 87, /* sync */
010010000    , 88, /*  */
010010001    , 89, /*  */
010010010    , 90, /*  */
010010100    , 91, /*  */
010010101    , 92, /*  */
010010110    , 93, /*  */
010011000    , 94, /*  */
010011001    , 95, /*  */
010011010    , 96, /*  */
010011110    , 97, /* sync */
010100000    , 98, /*  */
010100001    , 99, /*  */
010100010    ,100, /*  */
010100100    ,101, /*  */
010100101    ,102, /*  */
010100110    ,103, /*  */
010101000    ,104, /*  */
010101001    ,105, /*  */
010101010    ,106, /*  */
010101100    ,107, /*  */
010101101    ,108, /*  */
010101110    ,109, /* sync */
010110000    ,110, /*  */
010110001    ,111, /*  */
010110010    ,112, /*  */
010110100    ,113, /*  */
010110101    ,114, /*  */
010110110    ,115, /*  */
010111110    ,116, /* sync */
011000000    ,117, /*  */
011000001    ,118, /*  */
011000010    ,119, /*  */
011000100    ,120, /*  */
011000101    ,121, /*  */
011000110    ,122, /*  */
011001000    ,123, /*  */
011001001    ,124, /*  */
011001010    ,125, /*  */
011001100    ,126, /*  */
```

[Drawing 8]

```
011001101    ,127,/*  */
011001110    ,128,/* sync */
011010000    ,129,/*  */
011010001    ,130,/*  */
011010010    ,131,/*  */
011010100    ,132,/*  */
011010101    ,133,/*  */
011010110    ,134,/*  */
011011000    ,135,/*  */
011011001    ,136,/*  */
011011010    ,137,/*  */
011011110    ,138,/* sync */
011111110    ,139,/* sync */
100000000    ,140,/*  */
100000001    ,141,/*  */
100000010    ,142,/*  */
100000100    ,143,/*  */
100000101    ,144,/*  */
100000110    ,145,/*  */
100001000    ,146,/*  */
100001001    ,147,/*  */
100001010    ,148,/*  */
100001100    ,149,/*  */
100001101    ,150,/*  */
100001110    ,151,/* sync */
100010000    ,152,/*  */
100010001    ,153,/*  */
100010010    ,154,/*  */
100010100    ,155,/*  */
100010101    ,156,/*  */
100010110    ,157,/*  */
100011000    ,158,/*  */
100011001    ,159,/*  */
100011010    ,160,/*  */
100011110    ,161,/* sync */
100100000    ,162,/*  */
100100001    ,163,/*  */
100100010    ,164,/*  */
100100100    ,165,/*  */
100100101    ,166,/*  */
100100110    ,167,/*  */
100101000    ,168,/*  */
100101001    ,169,/*  */
100101010    ,170,/*  */
100101100    ,171,/*  */
100101101    ,172,/*  */
100101110    ,173,/* sync. */
100110000    ,174,/*  */
100110001    ,175,/*  */
100110010    ,176,/*  */
100110100    ,177,/*  */
100110101    ,178,/*  */
100110110    ,179,/*  */
100111110    ,180,/* sync */
101000000    ,181,/*  */
101000001    ,182,/*  */
101000010    ,183,/*  */
101000100    ,184,/*  */
101000101    ,185,/*  */
101000110    ,186,/*  */
101001000    ,187,/*  */
101001001    ,188,/*  */
101001010    ,189,/*  */
101001100    ,190,/*  */
101001101    ,191,/*  */
101001110    ,192,/* sync */
```

[Drawing 9]

```
101010000       , 193, /*  */
101010001       , 194, /*  */
101010010       , 195, /*  */
101010100       , 196, /*  */
101010101       , 197, /*  */
101010110       , 198, /*  */
101011000       , 199, /*  */
101011001       , 200, /*  */
101011010       , 201, /*  */
101011110       , 202, /* sync */
101100000       , 203, /*  */
101100001       , 204, /*  */
101100010       , 205, /*  */
101100100       , 206, /*  */
101100101       , 207, /*  */
101100110       , 208, /*  */
101101000       , 209, /*  */
101101001       , 210, /*  */
101101010       , 211, /*  */
101101100       , 212, /*  */
101101101       , 213, /*  */
101101110       , 214, /* sync */
101111110       , 215, /* sync */
110000000       , 216, /*  */
110000001       , 217, /*  */
110000010       , 218, /*  */
110000100       , 219, /*  */
110000101       , 220, /*  */
110000110       , 221, /*  */
110001000       , 222, /*  */
110001001       , 223, /*  */
110001010       , 224, /*  */
110001100       , 225, /*  */
110001101       , 226, /*  */
110001110       , 227, /* sync */
110010000       , 228, /*  */
110010001       , 229, /*  */
110010010       , 230, /*  */
110010100       , 231, /*  */
110010101       , 232, /*  */
110010110       , 233, /*  */
110011000       , 234, /*  */
110011001       , 235, /*  */
110011010       , 236, /* not used */
110011110       , 237, /* not used */
110100000       , 238, /* not used */
110100001       , 239, /* not used */
110100010       , 240, /* not used */
110100100       , 241, /* not used */
110100101       , 242, /* not used */
110100110       , 243, /* not used */
110101000       , 244, /* not used */
110101001       , 245, /* not used */
110101010       , 246, /* not used */
110101100       , 247, /* not used */
110101101       , 248, /* not used */
110101110       , 249, /* not used */
110110000       , 250, /* not used */
110110001       , 251, /* not used */
110110010       , 252, /* not used */
110110100       , 253, /* not used */
110111110       , 254, /* not used */
111111110       , 255, /* not used */
```

[Drawing 11]

RunL	Amplitude 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	2	3	4	5	5	6	6	6	7	7	7	8	8	8	9	9
1	3	5	7	8	9	9	10	11	11	11	12	12	13	13	13	13
2	5	7	9	10	11	12	13	14	15	16	16	16	16	17	17	18
3	5	8	10	12	13	14	15	16	16	18	17	20	20	18	17	18
4	6	9	11	13	15	16	17	19	20	19	20	19	20	20	20	19
5	6	10	12	14	15	15	16	17	19	17	20	18	19	18	19	19
6	6	10	13	14	15	18	17	17	17	18	18	19	18	19	19	19
7	7	11	13	14	15	16	18	18	19	18	18	18	18	19	18	18
8	8	11	14	15	17	18	18	19	18	19	19	19	19	19	19	19
9	8	11	14	17	16	17	18	19	18	19	19	19	19	19	19	19
10	8	12	14	16	16	18	17	17	19	19	19	19	19	19	19	19
11	9	13	15	16	17	16	16	17	19	19	19	18	19	18	19	18
12	9	13	15	19	19	19	19	19	19	19	19	19	19	19	19	19
13	10	14	16	19	18	19	19	19	19	19	19	19	19	19	19	19
14	10	15	16	19	19	19	19	19	19	19	19	19	19	19	19	19
15	11	16	18	19	19	19	19	19	19	19	19	19	19	19	19	19

[Translation done.]

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is the block diagram of the record data-processing circuit of the digital video tape recorder with which this invention was applied.
[Drawing 2] It is the approximate line Fig. showing bit assignment of the two-dimensional Huffman table by this invention.
[Drawing 3] It is the approximate line Fig. showing an example of bit assignment.
[Drawing 4] It is the approximate line Fig. showing an example of bit assignment.
[Drawing 5] It is the approximate line Fig. of the code-conversion table for direct coding.
[Drawing 6] It is the approximate line Fig. of the code-conversion table for direct coding.
[Drawing 7] It is the approximate line Fig. of the code-conversion table for direct coding.
[Drawing 8] It is the approximate line Fig. of the code-conversion table for direct coding.
[Drawing 9] It is the approximate line Fig. of the code-conversion table for direct coding.
[Drawing 10] It is the approximate line Fig. of a JIGUZAKU scan of DCT multiplier data.
[Drawing 11] It is the approximate line Fig. showing bit assignment of the two-dimensional Huffman table proposed previously.
[Description of Notations]
4 DCT Circuit
7 Quantization Circuit
9 Amount-of-Data Estimated Machine
12 Two-dimensional Huffman Table

[Translation done.]

* NOTICES *

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Industrial Application] This invention relates to the two-dimensional Huffman coding approach applied to the multiplier data generated in DCT.
[0002]
[Description of the Prior Art] The digital video tape recorder which records a digital video signal on a magnetic tape by the rotary head is known. Since there is much amount of information of a digital video signal, high efficiency coding for compressing the transmission amount of data is adopted in many cases. Utilization of DCT (Discrete Cosine Transform) is progressing also in various high efficiency coding.
[0003] DCT changes the image of one frame into the block structure, and is a thing which is a kind of orthogonal transformation about this block and which carries out cosine transform processing (8x8). Consequently, the multiplier data of (8x8) are generated. The alternating current component of this multiplier data is outputted by JIGUZAKU scan in order toward a high region from low-pass, as shown in drawing 10 . In one direction flowed data (DC) are transmitted as it is, without carrying out variable length coding. After processing quantization, variable length coding of such multiplier data is carried out. Two-dimensional Huffman coding is known as one of the variable length coding. This carries out the lookup of the Huffman table in the group which consists of a run length (the number of continuation of a bit "0"), and amplitude of the multiplier following it, and generates a predetermined codeword.
[0004] Drawing 11 shows bit assignment of an example of a two-dimensional Huffman table. In drawing 6 , it is the amplitude of 1-16 horizontally, and is the run length of 0-15 in front of the amplitude perpendicularly. The maximum number of bits of the codeword in a table is 20 bits. if -- this table -- not corresponding (a run length, amplitude) -- when inputted, it is processed by the escape sequence. That is, such an input is encoded by total of the 22-bit code which consists of 6 bits which shows an escape code (6 bits) and a run length, 9 bits which shows the amplitude, and a sign binary digit.
[0005]
[Problem(s) to be Solved by the Invention] Since the maximum number of bits (an above-mentioned example 22-bit width of face) is required as a bus line, the direction with little maximum number of bits is desirable. Moreover, in the conventional Huffman code, when an error occurs on the way, there is a problem of a propagated error it becomes impossible for all the codes after it to decode.
[0006] Therefore, one purpose of this invention is to offer the Huffman coding approach and equipment with as little longest number of bits as possible, without reducing the effectiveness of a Huffman code.
[0007] Other purposes of this invention are to offer the Huffman coding approach and equipment in which a self-synchronization is possible by the big probability at the time of error generating.
[0008]
[Means for Solving the Problem] In the two-dimensional Huffman coding approach that this invention generates a variable-length codeword with reference to a table from the group of the amplitude of input

data, and a run length Run length = It is made to perform the 1st escape sequence in 0, and the 2nd escape sequence in case a run length is except zero alternatively. The total bit length m by which the total bit length n generated by the 1st escape sequence is generated by nothing and the 2nd escape below the maximum bit length in a table is longer than the maximum bit length, and is the two-dimensional Huffman coding approach characterized by the thing [being made smaller than twice].

[0009]

[Function] The maximum bit length of the codeword specified on a table is shortened more. In that case, it is not specified on a table but the range where an escape sequence is applied spreads. On the other hand, decline in effectiveness can be prevented by preparing the 1st and 2nd escape sequences.

[0010]

[Example] Hereafter, one example of this invention is explained with reference to a drawing. Drawing 1 shows the configuration of the processing circuit of a video data established in the record side of a digital video tape recorder. In drawing 1 , the digitized video data is supplied to the input terminal shown by 1. This video data is supplied to the blocking circuit 2. The video data of the sequence of a raster scan is changed into the data of the structure of a DCT block of a for example, (8x8) in the blocking circuit 2.

[0011] The output of the blocking circuit 2 is supplied to the shuffling circuit 3. In the shuffling circuit 3, the processing which changes a spatial location with the original thing, i.e., shuffling, is made by making two or more macro blocks into a unit within one frame so that an error may concentrate and it may prevent that degradation of image quality is conspicuous with a drop out, the blemish of a tape, a head clog, etc. In this example, it is equal and the shuffling unit and the buffering unit are considered as 5 macro block. The output of the shuffling circuit 3 is supplied to the DCT (cosine conversion) circuit 4 and the motion detector 5. From the DCT circuit 4, the multiplier data (namely, multiplier data of the direct-current part DC and an alternating component AC) of (8x8) are generated.

[0012] It is transmitted to a latter circuit, without compressing the direct-current part DC of the multiplier data generated in the DCT circuit 4 (8x8), and 63 of the alternating components AC1-AC63 of it are supplied to the quantization circuit 7 through a buffer 6. The multiplier data of an alternating component are transmitted in order as mentioned above toward what has this high in order of a JIGUZAKU scan from an alternating component with a low degree. Moreover, the multiplier data of this alternating component are supplied also to the AKUTI Beatty detector 8 and the amount-of-data estimated machine 9. The buffer 6 has time amount required to determine the suitable quantization number QNo with the estimated vessel 9, and the corresponding amount of delay. The quantization number QNo from the estimated machine 9 is transmitted to the latter part while it is supplied to the quantization circuit 6.

[0013] Generating of the multiplier data from the above-mentioned DCT circuit 4 is the case of the DCT conversion in a frame, and if there is a motion and it will be detected by the motion detector 5, processing of DCT in the field will be chosen by it. That is, the inside DCT of the field changes every two DCT blocks of the same location in the 1st and 2nd fields which continue in time of (4x8). It will move, if there is a motion between the fields, and if a detector 5 detects, this detection will be answered and it will be changed into [DCT] the field. While the amount-of-data estimated machine 9 is presented with the detecting signal (motion flag) M from the motion detector 5, it is transmitted to the latter part.

[0014] The alternating component in multiplier data is quantized in the quantization circuit 7. That is, division of the multiplier data of an alternating component is carried out by the suitable quantization step, and the quotient is integer-ized. This quantization step estimates and it is determined by the quantization number QNo from a vessel 9. In the case of a digital video tape recorder, since processing of edit etc. is made per the 1 field or one frame, the 1 field or the amount of transaction datas per frame needs to become below desired value. Since the amount of data generated in DCT and variable length coding changes with the patterns of the object of coding, the buffering processing for making the amount of transaction datas of a buffering unit shorter than the 1 field or an one-frame period below into desired value is made. A buffering unit is shortened because [of simplification of a buffering circuit, such as reducing the memory space for buffering,].

[0015] Moreover, the AKUTI Beatty detector 8 is the unit of a DCT block, investigates the amount of an alternating current component and generates the 2-bit bitter taste tee BIITI code AT which shows a class part opium poppy and its class to four steps for AKUTI Beatty of the DCT block. A detection result estimates, a vessel 9 is supplied and the bitter taste tee BIITI code AT is transmitted to the latter part.
[0016] Variable length coding of the output of the quantization circuit 7 is supplied and carried out to the variable-length coding network 11. For example, the Huffman table in which the run length which is the number of continuation of "0" of the multiplier data of a code, and the amplitude of multiplier data were stored in ROM is given, and the two-dimensional Huffman coding which generates a variable length code (coding output) is adopted. The code signal from the variable-length coding network 11 is supplied to the latter part.
[0017] In relation to the estimated machine 9, the same Huffman table 12 as being referred to by the variable-length coding network 11 is formed. This Huffman table 12 generates the number-of-bits data of the output code when carrying out variable length coding. The group of the optimal quantization step is judged with the estimated vessel 9, and the decision output is supplied to a selector 10. A selector 10 is controlled so that the quantization circuit 7 quantizes multiplier data in the group of this quantization step. The quantization number QNo for identifying the group of a quantization step with this is transmitted to the latter part.
[0018] not illustrating, either -- in the latter part, the data (direct-current part data, a variable-length-coding output, the quantization number QNo, the motion flag M, the bitter taste tee BIITI code AT) generated in above-mentioned processing receive processing of error correction coding, and are further changed into the frame structure of record data in a frame-ized circuit. The data of a sink block configuration appear from a frame-ized circuit. Record data are supplied to two rotary heads through a channel coding network and record amplifier, and are recorded on a magnetic tape.
[0019] This invention is applied to the variable-length coding network 11 and the variable-length coding network of the amount-of-data estimated machine 9. More specifically, it is the configuration of the Huffman table 12. Drawing 2 shows bit assignment of the Huffman table on which this invention was applied. The horizontal direction of drawing 2 is the amplitude of multiplier data, and the perpendicular direction is a run length in front of multiplier data. Since this drawing 2 does not contain a sign binary digit, when this is added, the maximum number of bits is 16. The coding output of 92 pieces is specified in the table of drawing 2 . As for this field specified, the number of bits is specified in consideration of the probability of occurrence of the multiplier data after the above quantization. That is, since the probability of occurrence of low-pass multiplier data is high as compared with the thing by the side of a high region, the quota number of bits is made small.
[0020] Drawing 3 and drawing 4 show the example of the Huffman table. 95 codewords are prescribed by all on the table of drawing 3 and drawing 4 . In drawing 3 and drawing 4 , s expresses a sign binary digit, it is the special reserve bit by which R is added to the end of block (EOB), and code length does not contain a sign binary digit. Furthermore, the 92 remaining codewords except the 1st and 2nd below-mentioned escape codes ESC1 and ESC2 are contained in the table of drawing 2 .
[0021] In drawing 3 and drawing 4 , what was set to "sync" by the item of a type has 4 bits of "1110" for self-synchronizations as 4 bits of the last of a codeword. These four bit patterns are not generated except 4 bit of the last of one codeword. Therefore, if the error flag which shows generating of an error stands, propagation of an error can be cut off by looking for four bit patterns of "1110."
[0022] Next, an escape sequence is explained. This is divided into two sequences. One of them is run length =0, and it is the case of amplitude >16. In this case, a total of 16 bits of the 9 bits and the sign binary digit of escape code ESC1= "000110" (6 bits) and the amplitude encode. Coding of the amplitude is fixed-length direct coding. Run length = as for the data length generated in the escape sequence in 0, it is desirable that it is below the maximum bit length so that a transmission rate may not become high.
[0023] A run length is not 0 and, in besides the table of drawing 2 , the 2nd escape sequence is applied. This is 31 bits in total and is divided into a 15 bits segment and a 16-bit segment. The 1st segment is 15 bits which consists of escape code ESC2= "00010000" (8 bits) and run lengths (7 bits). The 2nd segment is 16 bits of the 9 bits and the sign binary digit of escape code ESC1= "000110" (6 bits) and the

amplitude in total. As for coding of a run length and the amplitude, fixed-length direct coding is applied. At the time of transmission, since it is divided into the 1st segment and 2nd segment, it can prevent that the bit length of data exceeds 16 bits substantially.
[0024] As a run length, since a maximum of 63 is assumed, at least 6 bits is required here, and as amplitude, since a maximum of 255 is assumed, at least 8 bits is required. Furthermore, in order to make it four bit patterns for an above-mentioned synchronization "1110" not generated, these are increasing 1 bit at a time, respectively. Drawing 5 , drawing 6 , drawing 7 , drawing 8 , and drawing 9 are the code-conversion tables of direct coding. According to these code-conversion tables, direct coding of the run length or amplitude besides a table is carried out.
[0025] This invention is applicable also to variable length coding of data word other than the multiplier data generated in DCT.
[0026]
[Effect of the Invention] According to this invention, the maximum bit length of the codeword generated in two-dimensional Huffman coding can be set to 16 (a sign binary digit is included). Therefore, hardware is made, such as lessening bit width of face of a bus run length, as it is simple. Moreover, this invention has the high probability for the bit pattern for a synchronization to be generated, and is easy a self-synchronization, consequently can shorten error propagation.

[Translation done.]

CLAIMS

[Claim(s)]
[Claim 1] In the two-dimensional Huffman coding approach which generates a variable-length codeword with reference to a table from the group of the amplitude of input data, and a run length Run length = It is made to perform the 1st escape sequence in 0, and the 2nd escape sequence in case a run length is except zero alternatively. The two-dimensional Huffman coding approach that the total bit length m by which the total bit length n generated by the 1st escape sequence of the above is generated by nothing and the 2nd above-mentioned escape below the maximum bit length in the above-mentioned table is longer than the above-mentioned maximum bit length, and is characterized by the thing [being made smaller than twice].
[Claim 2] The two-dimensional Huffman coding approach characterized by selecting to n= 15 and m= 31 when the above-mentioned maximum bit length is set to 16 in claim 1.
[Claim 3] The two-dimensional Huffman coding approach characterized by having two or more bits for a specific synchronization partially in the above-mentioned codeword in the two-dimensional Huffman coding approach which generates a variable-length codeword with reference to a table from the group of the amplitude of input data, and a run length.
[Claim 4] The two-dimensional Huffman coding approach characterized by selecting in the bit for [several bits] the above-mentioned synchronization of the low order of the above-mentioned codeword in claim 3.

[Translation done.]